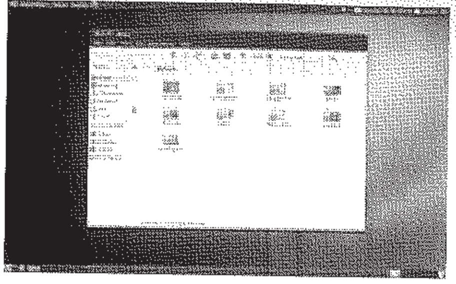
An **operating system** (or **OS**) is a group of computer programs, device drivers, kernels, and other software that let people interact with a computer. It manages computer hardware and software resources. It provides common services for computer programs. An OS can be small (like MenuetOS), or large (like Microsoft Windows). Different operating systems can be used for different purposes. Some are used for everyday things like on a personal computer. Others are mobile operating systems or are used for specialized work.


Ubuntu GNU/Linux, a free operating system

An operating system has many jobs. It makes sure that all the programs can use the CPU, system memory, displays, input devices, and other hardware. Some also give the user an interface to use a computer. An OS is also responsible for sending data to other computers or devices on a network.

Some examples of commonly used operating systems are macOS, Linux, and Microsoft Windows.

[illegible]

[illegible]

Contents

History

The first operating system was used with the ENIAC (Electronic Numerical Integrator and Computer).[1] It was very hard to make ENIAC do work. How the operating system worked was based on how the switches and cables were put together and depending on this factor punch cards would make a result. While this was an operating system of a kind, it is not what is thought of as one in modern times.

The first operating system that looked and felt like operating systems in the modern age was UNIX, made in 1969 by Bell Labs. It had a small kernel and many tiny programs that could be put together to work with user input and data. Many of its features were taken from Multics, an older operating system made in 1964.[1]

A **programming language** is a type of written language that tells computers what to do in order to work. Programming languages are used to make all the computer programs and computer software. A programming language is like a set of instructions that the computer follows to do something.

A programmer writes source code text in the programming language to create programs. Usually, the programming language uses real words for some of the commands, so that the language is easier for a human to read. Many programming languages use punctuation just like a normal language. Many programs now are "compiled". This means that the computer translates the source code into another language (such as assembly language or machine language), which is much faster and easier for the computer to read, but much harder for a person to read.

Computer programs must be written very carefully. If the programmer makes mistakes, or the program tries to do something the programmer did not design it to do, the program might then "crash" or stop working. When a program has a problem because of how the code was written, this is called a "bug". A very small mistake can cause a very big problem.

Contents

[illegible]

Types of programming languages

There are many types of programming languages. Most programming languages do not follow one type alone, so it is difficult to assign a type for each language. The examples of each type are given in each section below because they are the best well-known examples of that type.

High-level vs. low-level

High-level programming languages require less knowledge about the hardware compared to low-level programming languages. High-level programming languages require an interpreter to convert the source code into low-level programming languages.

Declarative vs. Imperative programming

Declarative programming languages describe a "problem" but they usually do not say how the problem should be solved. The problem description uses logic, and "solving" the problem often looks like automatically proving a system of logical axioms. Examples for such programming languages are Prolog, XSLT, LISP and SQL.